DOCUMENT RESUME

ED 244 545 HE 017 248

AUTHOR Rokicki, Phillip S.; Springer, Peggy
TITLE Tying Diverse Student and Administrative Data Bases Together.
PUB DATE Dec 83
NOTE 13p.; In: Information Resources and the Individual. Proceedings of the CAUSE National Conference (San Francisco, CA, December 11-14, 1983). Boulder, CO, CAUSE, 1984, p. 53-64. For the complete proceedings, see HE 017 245.
PUB TYPE Viewpoints (120) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Administrator Guides; *College Administration; *Computer Oriented Programs; *Databases; Documentation; Higher Education; *Information Networks; *Management Information Systems; *Student Characteristics
IDENTIFIERS University of Missouri

ABSTRACT

Guidelines for colleges seeking to interface their diverse student and administrative databases are presented. The following key concepts are addressed that need to be understood before solving the problem of sharing and transmitting data between diverse databases: effective management of data from an institutional and a user point of view; the problems encountered by ownership of data; and the redundancy and duplication already existing in databases. As part of researching the data-sharing problem, the policymaker must also understand what is the proper blend of technology for that institution and what are the politics of that proper blend. Besides the institution's own design for managing data, the project will require a conceptual plan to determine what data is to be shared and how the sharing is to be accomplished. While the systems life cycle can be applied to a project involving only a single database, it may also be utilized to link separate databases. The cycle has five separate stages: definition phase, design phase, development phase, test phase, and implementation. Attention is also directed to the development of documentation and the specific experiences of the University of Missouri in tying databases together. (SW)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED244545

TYING DIVERSE STUDENT AND ADMINISTRATIVE DATA BASES TOGETHER

by
Dr. Phillip S. Rokicki, Director
Scholarships and Student Financial Aid Services
University of Missouri
Columbia, Missouri

and

Ms. Peggy Springer, Project Manager
Computing and Information Systems
University of Missouri
Columbia, Missouri

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

CAUSE

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Postsecondary education, over the past several decades, has instituted automated data bases for very specific reasons. Student-oriented data bases were developed for admissions and records, for housing, for payment of fees and other receivables, and lately for the handling of student financial aid. These are in addition to administrative data bases for payroll, accounting, and accounts payable. All of these data bases, originally independent of each other, carry at least some data which should be available to all segments of the institution.

Redundancy and duplication of data exists in these data bases. The administrative costs associated with maintaining those separate data systems are often unnecessary, and adequate reports are almost impossible to secure. Technology is not being used to its best advantage; rather, the fragmentation caused by multiple data bases is working against cost-effective administration.

To solve the problems of tying separate data bases together, an institution must consciously seek to solve the political, emotional, and technical issues involved. Not only is adequate planning required, but cooperation coupled with strong administrative support is necessary. Institutions undertaking the interfacing of separate data bases need to understand the various pitfalls and road blocks before writing a single line of interface code. This paper offers general guidelines to institutions seeking to interface together their diverse student and administrative data bases.

Paper presented at the
CAUSE National Conference
(San Francisco, CA, Dec. 11-14, 1983)

HE 017 248

Computerization in American postsecondary education has turned a corner. No longer do we talk about the wisdom of whether or not to computerize or even to research the benefits of computerization. Today, with the proliferation of data bases not only on institutional mainframes but also now on mini- and micro-computers, we are beginning to attempt to tie diverse data bases together. While our discussion will center on the area of tying student and adminstrative data bases together, the problems and challenges encountered in this area can be generalized to other areas of the institution where more than a single data base exists.

Those of you contemplating the attempt to tie diverse data bases together should remember a short poem by Lewis Carroll entitled, "How Doth the Little Crocodile."

How doth the little crocodile
Improve his shining tail,
And pour the waters of the Nile
On every golden scale!

How cheerfully he seems to grin,
How neatly he spreads his claws
And welcomes little fishes in
With gently smiling jaws!

Those who seek to bring order out of data base turmoil will be much like Carroll's little fishes swimming into the smiling jaws of the crocodile. But fear not; those teeth may be avoided, but it will take planning, understanding, and a fair amount of suffering.

A. Introduction to the Problem

Before you or I can begin to solve the problem of sharing and transmitting data between diverse data bases, we first need to understand several key concepts -- the effective management of data from an institutional and a user point of view, the problems encountered by ownership of data, and the redundancy and duplication already exisiting in data bases. Before we begin to examine data base problems, let's explore these three areas.

1. Effective Management of Data from the Perspectives of the Institution and User

In a fantasy world, everyone in an institution (or for that matter, within even a single office) would completely share appropriate data with co-workers. Instead of "War Games" movies, we would find intra-institutional cooperation at a very high level. We would all work together for a perceived common good.

Obviously, this fantasy world is not the real world of institutions today. Individuals perceive the ownership of data as power which is rightfully theirs to dispense. Your deadlines and priorities often are meaningless to others outside your immediate sphere of influence.

In a much broader sense, what is determined to be good management of data for an institution may not be in the best interest of a user. Likewise, what may be perceived as good management for an office may not be in the best interest of the institution and/or user. Here we shall stop to define the term "user." We will use the term to describe any individual or group who

collects, records or manipulates data. NOTE: This data may or may not be machine retrievable; it can be stored on a mainframe or on 3 X 5 cards.

Deciding how you will manage data for an institution is the first step in bringing a rational approach to the sharing of data. Before any institution should undertake the controlling of data, it should determine the rules of the game for institutional data management. Some of the key questions which an institution must ask itself are:

- What is the scope of data collected by this institution and in what form is the data stored?
- Which groups or individuals are responsible for the integrity of the data?
- Is it important to the institution that data be consistent when shared internally and externally?
- How accurate must this data be for it to be useful?

Only after an institution begins to understand its current data structure can it begin to undertake the appropriate sharing of data within the institution.

As part of this institutional data soul-searching, the role of the various users will become clear. It is the user, after all, who initiates the collection of data, records the data in some way, and communicates that data. The institution, as part of its own understanding of the data universe, must ask the users questions such as:

- Why is the data collected and how is it used?
- How is the data stored?
- Who else outside the primary user accesses the data and how often?
- Are there others who could utilize the data?

Once the institutional managers of data have completed a rather comprehensive review, not only of how the institution expects to manage data, but how users assist or hinder in that management of data, then planning may proceed.

2. Problems of Ownership of Data

Very early in any institution's data soul-searching process, the manager may find problems with one or more individuals who feel they "own" data. This feeling of ownership is not limited to office managers, it also includes clerks and secretaries who feel a certain sense of power because they control data. How many of you have found out that a rather seemingly trivial task cannot be carried out because someone is sick or on vacation? (How many decisions will have to wait for your return because you "own" certain data?)

You must not only recognize the problem of data ownership, but you must assist those who have that problem. You can begin to resolve the problem in a number of ways, some of which include:

a. Hold discussions with those who currently "own" the data. Reassure them that the loss of ownership will not result in layoffs or loss of authority, but will result in increased responsibility in the data-sharing process.

b. Involve the data owners in problem-solving sessions whereby alternate ways of collecting and recording data are investigated and the needs of others to have access to the data are explained.

c. If all else fails, specific instructions from senior institutional officers that data is to be shared may be necessary, although this is the least effective way to encourage cooperation.

No single approach works every time for everyone involved with data ownership. Rather, the approach used with one user may involve many sub-approaches. It is quite clear, though, that ignoring the data ownership question in the early stages of the process will cause significant disruption later.

3. Redundancy and Duplication of Data Within the Institution

As part of its institutional review of data management, the managers must begin to understand the redundancy and duplication which already exists within the institution. While it is impossible for us to cover all possible examples of redundancy and duplication, a good example will aid you in starting to understand the scope of the problem. Let's use the very basic student demographic data of name, address, and phone number. If you answer the following questions you may be surprised at the bureaucratic tangle we put students and ourselves through for three such seemingly simple items.

a. How many offices collect that data from prospective students and current students? In institutions without data base sharing, a student's address would have to be changed, manually or via data entry, in at least the following offices:

1) Admissions
2) Registrar
3) Financial Aid
4) Housing
5) Parking
6) Cashier/Business
7) Student Activities/Dean of Students
8) Student's selected college or school
9) Student's major or minor department
10) Development/Alumni

b. How many forms does a student have to complete to change one of the items?

c. How long does it take to notify others who use the data that a change has occurred?

d. How many people are involved with the changing of data, assuming all possible offices and departments receive notice of the change?

e. How long does it take all affected offices to completely change this data for a single student?

While a particular institution may have a longer or shorter list of involved offices, it is quite clear: **simple data is the hardest to keep current and requires the most manual effort to transmit and to maintain.** Additional types of demographic data (including grades, enrollment status, and graduation information) would each have its own list of users who need or think they need the data. While some data is only of interest to a single office, most student data is of interest to more than one office.

Redundancy of data increases costs, in both time and effort, for those who have to complete forms or maintain the data. It also causes excess data management costs for the institution. Whether those costs be for on-line or off-line storage or file folders and 3 X 5 cards, it is a duplication of "costs" in personnel time and material. In addition, reporting accuracy and consistency are most difficult to achieve with such a system. However, there are some institutionally defined legitimate reasons for the existence of duplicate data (e.g., operation of vendor written software). However, **the institution should eliminate the duplication of the updating of this data.**

B. Politics and Technology — Must Conflict Exist?

Technology, as we know it, is neither good nor evil in itself; rather, individuals give technology its "taint" of black or white. Likewise, politics can work for the betterment of the institution or can work against the institution. As part of researching the data-sharing problem, every policy maker must understand what is the proper blend of technology for that institution and what are the politics of that proper blend.

It is impossible to prescribe for every institution what level of technology is appropriate. Obviously, what is correct for one institution may be too little or too much for another institution. Rather, what we suggest - - no, we urge - - is that the institution develop a road map towards the appropriate usage of modern technology to solve the defined problem. As part of that planning, the institution must understand the politics which will foster or hinder that development. Politics here means a variety of things, including:

- Once the appropriate level of technology is determined to solve the problem, what resources are available to secure and support that level of technology?
- What is the level of acceptance/resistance to the achievement of the determined level of technology?
- To achieve the desired level of development, how much real support exists among the board of control, senior institutional administrators, middle- and lower-level managers, and support staff?

If the level of support both in dollars and in acceptability is not readily apparent, then additional support must be achieved in both areas before the data sharing project is undertaken. Attaining a higher level of technology will require a certain expenditure of human and financial resources; therefore, the institutional political attitude must be correct. Without the proper amount of both, setbacks, disillusionment, and failure probably will result. It is better not to undertake a project or to scale down the project before it is begun, rather than to face doubtful success.

C. Attacking the Problem

When any institution begins to attack the problem of sharing data, a number of postulates must be understood. They are:

- "Might" doesn't always make it work.
- Cooperation has pitfalls.
- Personal energy will only go so far.

"Might" or administrative support, even from the highest of the institutional offices, will not, in itself, make the project succeed. Without high level support it is doubtful the project can succeed; don't expect an edict from the President or Chancellor to be your ultimate weapon on which success rides.

Cooperation does have its own pitfalls. Since you must rely on others for their help, you can be lulled into a false sense of security early in the project. While at first people may seem to be cooperative, as the project continues on, some of these people may become threatened with the loss of power or by the heavy burden of labor required of them. Therefore, the managers of the project must be constantly monitoring the willingness of those involved in the project for items such as missed deadlines, the constant pointing out of problems (real or imagined), and over-extended caution by participants.

While it is necessary to have both the administrative support of "might" and the cooperative attitude of staff, it is equally important to have the personal dedication of a staff to carry out the project. This dedication must be understood both in human and emotional terms. While a committee may be charged with the responsibility of making the project succeed, our experience indicates that usually two or three people really do all the meaningful work.

The project will require the key staff to be tactful politicians and skilled technocrats who pick up where others fall short or fear to tread. It will require these individuals to convince, to lobby, and to work over-time to achieve the goal of the project. **Freedom from other duties for these key people is one additional price the institution must pay for success.**

D. The Role of Planning

A project of this magnitude, interfacing diverse data bases, cannot be undertaken without considerable planning at various abstract levels. Besides the institution's own design for managing data, the project will require a conceptual plan to determine what data is to be shared and how the sharing is to be accomplished. An integral part of this plan must be the institution's policy on security and privacy as it relates to computing.

The primary conceptual plan should be as inclusive as possible, utilizing the systems life cycle model or a similar planning technique. The conceptual model must encompass as many data source areas as possible. As new data areas are "discovered," the conceptual map should be modified. Upon conclusion of the project, the plan then reflects the project "as built" rather than as originally planned or conceived.

While the systems life cycle can be applied to a project involving only a single data base, it may also be utilized to link separate data bases. The cycle has five separate stages:

1. **Definition Phase** - In this phase you define in detail what you are trying to do with the system or process. You explain which parts of the process are already automated and to what extent, and which parts are currently done manually. As an important part of the definition phase, you explain who is involved with the process and to what extent. The definition phase is the most **critical** of all the phases because it is the foundation on which the entire remainder of the project is built.

   The definition phase normally will include statements on the cost of the current process and cost of the proposed process with any savings (or future cost avoidance) noted. Responsibilities are usually defined in the document and success is defined even before the process begins.

2. **Design Phase** - This phase builds on the information gathered in the definition phase. It refines the definition phase, and the proposed processes are written in more enhanced detail. Processes which will be changed are defined. As well, new pathways for moving information or data are determined. This phase must be completed before a single line of code is written, otherwise it is probable that significant errors may be made in the project. The design phase does not ensure the absence of all errors, rather it minimizes the number and probably the severity of the errors.

   As part of the Design Phase, the team must consider the consequences of the following:

   a. Real time update versus a batch update. Here you must consider just how important it is that data be 100% accurate at any given point in time. You should also decide over what time frame this degree of accuracy is required (e.g., twelve months or only on registration days).

   b. How necessary is it that the primary or secondary sources for data update their information immediately or every week, etc? You need to decide what are the critical data elements in the Design Phase and then find out just how often they are updated today. (You may be unpleasantly surprised with what you learn.)

   c. What will happen if one of the data bases is not available at "interface" time, especially if it is a real-time interface?

3. **Development Phase** - This phase involves either internally producing software or purchasing software from outside vendors. In the process of tying two or more data bases together, the development phase is often complicated by the following:

   - The data bases have incompatible file structures.
   - The data bases have different keys (e.g., name versus student number versus social security number).
   - Data is defined differently in the data bases.

   During the development phase, most such problems will be identified as the team attempts to move data from one system to another system.

   If you have not up to this point reached the paradox of which system has the best data, it will confront you now. Obviously, the reason why you share data is to be able to have the **best** data available, but getting all the players in the game to agree on which data is the best (or who is the primary source for the data and who is supplementary) is difficult.

   During development, you must answer the following questions for each item of data:

   a. What is the best source of the data? The second best, etc.?

   b. If the data element is blank, who can provide the information?

   c. If a secondary source initially provides the data and the primary source can now provide the data, should the primary source overlay the information from the secondary source?

   d. Should the data field be protected so only one data provider can update or modify the field?

   e. What happens to unprotected fields if more than one data provider updates the field (e.g., one office changes the address and the next day someone else changes it back)?

   f. Should the data field be unprotected for parts of the year and protected for other parts of the year?

   g. Is everyone defining the data field the same way? An example would be the term enrolled - it probably means something entirely different to the admissions, financial aid, registrar, and business offices. You need to write a definition of each data element before you attempt to write your interface. Have all the game players agreed **in advance** on the meanings (overt and covert) of the data field?

   h. How will you recover your data bases if one or more is damaged? What will happen if your major student file is unavailable for a day? Who will inform other users as to what they can or cannot do that day? How often will you back up on-line or batch processes?

The writing of the specifications for the interface will take considerably longer than will the actual coding.

4. **Test Phase** - This is the time to try to operate the interfaces in conjunction with the data bases in a test environment. Output reports should be shared with the team members and with the users. The testing procedures should be substantial enough to attempt to simulate the system's entire cycle. The testing phase should not be minimized. **Before putting an interface in production, we recommend a signed agreement from each of the team players/users that they have adequately tested the interface and are satisfied with its results.**

5. **Implementation** - This is really more than just putting an interface into production, it is a close monitoring of the interface for an extended period of time. Problems will arise and the team should be ready to deal with them. Depending on the severity of the problem, caution should be exercised so that any quick fixes do not complicate the problem even further. Fixes should go through the same test procedures as did the original interface. Once again, agreement should be reached before the fixes are implemented into the system.

E. **Development of Documentation**

As is obvious from our discussion concerning the system life cycle, the tying of data bases requires more written work than actual programming effort. In providing documentation for the whole, you should consider including the following:

1. Statement on what you are attempting to do and why
2. A data flow diagram
3. Your Design Phase document
4. Rules for the movement of data as "discovered" or identified in the Design and Development Phases
5. A data element dictionary or similar documentation
6. Suggested job streams and job control language
7. Critical scheduling information
8. Sample output
9. Coding forms, sample on-line screens, etc.
10. All error messages, and an explanation (including cause and resolution, if not obvious) of each.

The documentation should be of sufficient depth to be used for 1) the introduction of senior institutional administrators to the interface. 2) the training of current and new office staff, and 3) technical information for data processing staff. Obviously, the material necessary for a data entry clerk may be different from that given to an institutional president, but we recommend an overview of sufficient

depth so either the clerk or the president is aware of the extent of the interface process. We also strongly recommend the ultimate user be involved in the development of the documentation.

F. Experiences of the University of Missouri in Tying Data Bases Together

Our presentation gives you tips on what to do or what to expect in tying data bases together. The reason we began tying data bases together was to solve a very difficult situation. We had, and still have, various offices which collect the same data from students, parents, and others. These offices report this data to the local press, the state government and federal agencies. It was very difficult to explain to the University President and campus Chancellors why similar data collected by various offices was almost always different.

We also became convinced we could eliminate students standing in line to repeat the same mechanical function by coordinating the updating of data. The vehicle which caused us to undertake this process in the first place was the purchase of a vendor produced software system for student financial aid.

Our student aid system is a gargantuan consumer of data. It feeds on data from other outside vendors, students, parents, other offices and data bases. We immediately faced a quandary - significant amounts of data are necessary for the process to begin and we have two ways to input the data - by hand or by some automated interface. If Missouri is similar to your institution, we are experiencing a financial crisis not expected to go away very soon. Therefore, we decided to undertake the coupling of data bases as the most economic of our choices, as well as the most timely and reliable. Currently all of our interfaces are batch processes.

We began with the most obvious interface - - that with our master student information system. This data base is managed a little differently by each campus, even though the data elements are defined consistently. We began by sitting back and deciding what data we needed, when it was available, and whether or not the data from our student information system was better or more accurate than our data in the student financial aid system. As you can imagine, the rules for this interface constitute a sizable document. Even after we were finished with testing and were in production, we still discovered errors, omissions, and misunderstandings.

If we learned anything with the first interface, it would be that no interface can be tested enough nor are the rules of the interface ever understood completely by all the players. Also, no interface remains completely static. We are constantly monitoring our interfaces watching for changes in the way data is inputted and changes in the way data is used.

Our second interface was with our Accounts Payable System, or the system to write checks. We undertook this interface to eliminate the hand typing of vouchers (or lists of requests for checks). This interface was really quite easy since the rules for the interface were established by the fiscal side, and they were rather stable. Transactions had to be in a certain format and anything else was rejected. Our only decision was what types of aid should go through the interface; and that debate still continues. In the future we expect to replace this interface with an interface to the Student Accounts Receivable System.

Our third interface seemed like a "piece of cake" - - it was to move College Work-Study earnings and non-federal student earnings into our student financial aid system. It seemed rather straightforward: our payroll system had its data in one form, it created payroll files at a set interval; therefore, we could easily read those files, match our student aid applicants up and transfer the data. Technically it is easy, but politically we opened a can of worms: never before did we have a realistic accounting of which student aid recipients were working for the University and how many dollars were involved. We did not know the size of the problem until we had written the interface. Now we know we have a number of student aid recipients who work for the University and who may be over-awarded.

A seemingly easy interface has opened a door that all agree needed to be opened. However, we were not prepared for what came out of the door. The simplistic design of the interface did not take into account the fact we were introducing significant amounts of new data into the system. We did not properly plan on handling this new data. We are now doing our retroactive homework with campus administrators on what the political effect will be on campus when you tell a student and a department that all on-campus work must stop for that student. Once again the simple interface has proved to be the most complex.

At the present time, we are in the process of beginning the following additional interfaces this year - an interface with the University's financial accounting system (general ledger) to report on actual expenditures for student aid in the area of other scholarships and fellowships, as well as any funds which have been "returned" to a given aid fund. We also expect to interface with our student loan system to create transactions it needs to set up long- and short-term loan borrowers. We are planning an interface with the housing system to accurately determine who lives on- or off-campus and a tentative automated feed to the Alumni system with information of interest for future development work.

We have found the interfacing of student and administrative data bases to be very interesting and challenging. As we interface more and more data systems, we find timing and cooperation most critical. Interfacing data systems solves some problems, but it creates a whole new set of challenges for the institution. We feel these new problems will and can be solved if we understand the people and processes used in moving data around an institution.

The University of Missouri system enrolls about 56,000 students on campuses in Columbia, Kansas City, Rolla, and St. Louis. Annually over 22,000 students receive about $80 million dollars in direct and indirect student financial aid. Over 33,000 students annually apply for financial aid.

G. **What Is Success**

In order to define success, an institution needs to evaluate its own "personality." What we consider success at the University of Missouri is probably different from what you consider success. Moving data around an institution manually is complex (as is evident by your campus mail system), and the complexity is magnified when you attempt to move the data electronically. You need to be realistic at the very first when you undertake the process of tying data bases together so you neither expect too much nor expect too little. However, it is better to be conservative in your statements of eventual success rather than to promise too much and be unable to deliver.

Even if one line of code is never written, the institution will benefit from having gone through the planning and evaluation exercise. The planning itself will bring to light areas which could and should share data.

The task to tie data bases together will require dedication and courage. We opened with a quote from Lewis Carroll, let's close with a quote from Jonathan Swift:

> "When a true genious appears in the world, you may know him by this sign: that the dunces are all in confederacy against him."

If you undertake the process, you may very well find quite a group in confederacy against you. However, you may encounter one of the most rewarding of all opportunities.

****